1. \*\*What is GraphQL, and how does it differ from REST?\*\*

GraphQL is a query language for APIs that allows clients to request only the data they need. Unlike REST, where the server determines the structure of the response, GraphQL enables clients to specify the shape and structure of the data they require, reducing over-fetching or under-fetching of data.

2. \*\*Explain the basic components of a GraphQL query.\*\*

A GraphQL query consists of three main components:

- \*\*Query\*\*: Describes the data requirements.

- \*\*Mutation\*\*: Specifies operations that modify data.

- \*\*Subscription\*\*: Enables real-time communication by establishing a connection between the client and server for data updates.

3. \*\*What is a resolver in GraphQL?\*\*

Resolvers are functions responsible for fetching the data for a specific field in a GraphQL schema. They define how to retrieve the data when a corresponding query is made. Resolvers are implemented for each field in a type.

4. \*\*Explain the concept of a GraphQL schema.\*\*

A GraphQL schema defines the types of data that can be queried and the relationships between them. It includes types for objects, queries, mutations, and subscriptions. The schema serves as a contract between the client and the server, specifying what data is available and how it can be requested.

5. \*\*How does GraphQL handle versioning?\*\*

GraphQL itself doesn't prescribe a specific versioning strategy, but it provides flexibility. Developers often version their GraphQL schemas by adding new types or fields without breaking existing queries. Backward-compatible changes are preferred to ensure existing clients continue to function.

6. \*\*What are the advantages of using GraphQL over REST?\*\*

- \*\*Efficient Data Retrieval\*\*: Clients can request only the data they need, reducing over-fetching.

- \*\*Single Endpoint\*\*: GraphQL typically uses a single endpoint, simplifying the API surface.

- \*\*Strongly Typed\*\*: GraphQL schemas provide a clear structure, and queries are validated against this schema.

- \*\*Real-time Updates\*\*: Subscriptions enable real-time data updates.

7. \*\*Explain the concept of batching in GraphQL.\*\*

Batching involves sending multiple GraphQL queries or mutations in a single request to reduce the number of network round-trips. This helps optimize performance by fetching multiple pieces of data in a single call, reducing latency.

8. \*\*How can you handle authentication in a GraphQL application?\*\*

Authentication in GraphQL is typically handled through HTTP headers, including tokens or other authentication credentials. These headers are sent with each GraphQL request. The server then verifies the credentials and processes the request accordingly.

9. \*\*What is the purpose of the introspection system in GraphQL?\*\*

The introspection system in GraphQL allows clients to query the schema itself. This means that clients can dynamically discover the types and capabilities of the API. It's useful for tools like GraphQL IDEs and documentation generators.

10. \*\*Explain the role of directives in GraphQL.\*\*

Directives in GraphQL provide a way to describe additional instructions to the execution engine. They can be used to conditionally include or skip fields, apply transformations to data, or control the execution flow based on certain conditions. Common directives include `@include` and `@skip`.